LISTING OF CLAIMS

1. (Currently Amended) A method for operating a computer system, said computer system including at least one processor, comprising:

establishing a plurality of memory units each having a corresponding memory location;

executing a plurality of tasks running on said at least one processor, said plurality of tasks being operable to share data;

defining a plurality of lists for each memory location;

determining the validity of said data in ~~said~~ one memory unit of said plurality of memory units;

locking at least one of said plurality of lists if said data is invalid;

inserting an entry corresponding to one of said plurality of tasks onto said locked list;

unlocking said locked list; and

determining if data is inputted in said corresponding memory location of said one memory unit between said determining step and said unlocking step.

2. (Original) A method for operating a computer system, said computer system comprising at least one processor, comprising:

establishing a plurality of memory units each having a corresponding memory location;

running a plurality of tasks on said processor, said plurality of tasks being operable to share data ;

defining a plurality of lists for each memory location;

inserting an entry corresponding to one of said plurality of tasks onto one of said plurality of lists if said one list is unlocked; and

determining if another of said lists is unlocked if said one list is locked.

3. (Currently Amended) A method for synchronizing processes in a computer system, said computer system including at least one processor, comprising:

establishing a plurality of memory units each having a corresponding memory location;

executing a plurality of tasks running on said processor, said plurality of tasks being operable to share data located in said memory units;

defining a plurality of lists for each memory location;

locking at least one of said plurality of lists if said data is not valid in one of said plurality of memory units;

inserting an entry corresponding to one of said plurality of tasks onto said locked list;

unlocking said locked list;

suspending said one of said plurality of tasks~~entered task~~ until valid data is found in said member of said memory units;

reading said valid data;

determining if other data is inputted in said one of said memory units ~~before~~ after said locking step and ~~after~~ before said unlocking step; and

reading said other data if said other data~~it~~ appears in said memory unit.

4. (Original) The method of claim 3, wherein the locking step further comprises activating selected other ones of said plurality of tasks that are entered on said locked list.

5. (Currently Amended) The method of claim 3, wherein said plurality of lists ~~forms~~ includes a linked list.

6. (Original) The method of claim 3, wherein said plurality of lists is between four and eight.

7. (Original) The method of claim 3, further comprising transferring the operation of said locked list when said locked list is locked by another one of said plurality of tasks.

8. (Original) A computer system having enhanced concurrency, comprising:

a plurality of processors;

a plurality of tasks running on said plurality of processors;

a plurality of memory units each having a corresponding memory location;

a plurality of lists corresponding to each of said memory location;

wherein one of said plurality of tasks is responsible for activating selected ones of said plurality of tasks contained on the same list as said one task.

9. (Original) The system of claim 8, wherein said plurality of lists form a linked list.

10. (Original) The system of claim 8, wherein said plurality of lists is between four and eight.

11. (Original) The system of claim 8, wherein said computer system is a multitasking or multiprocessing computer system.

12. (Currently Amended) A method of operating a computer system having at least one processor, comprising:

determining the validity of data in a memory unit;

locking a list corresponding to said memory unit if said data is invalid;

inserting an entry corresponding to one of ~~said~~ a plurality of tasks onto said locked list;

unlocking said locked list; and

determining if data is inputted in ~~a~~said memory location within said memory unit between said determining step and said unlocking step.